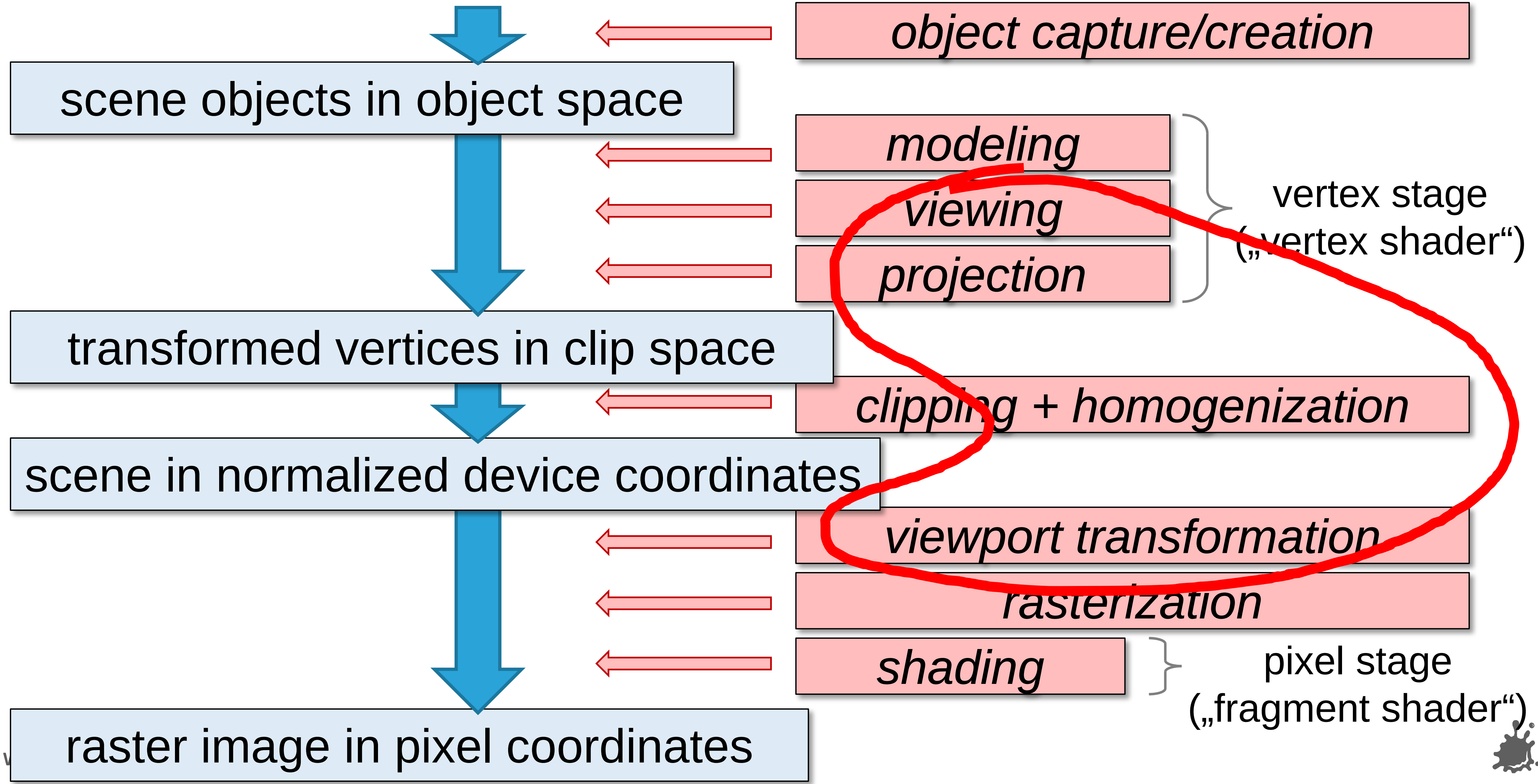
# Einführung in Visual Computing

186.822

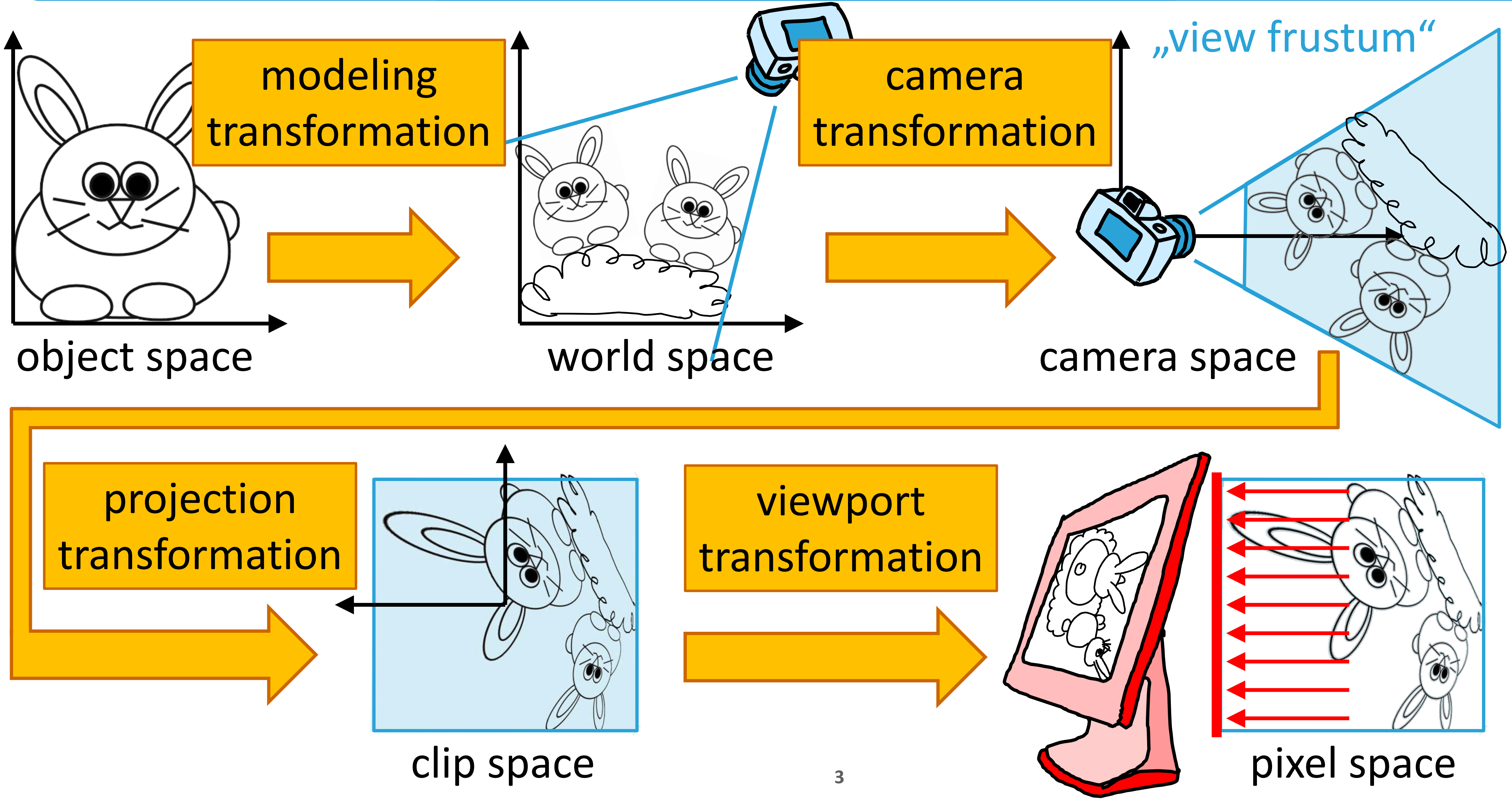
# Viewing

Werner Purgathofer

# Viewing in the Rendering Pipeline

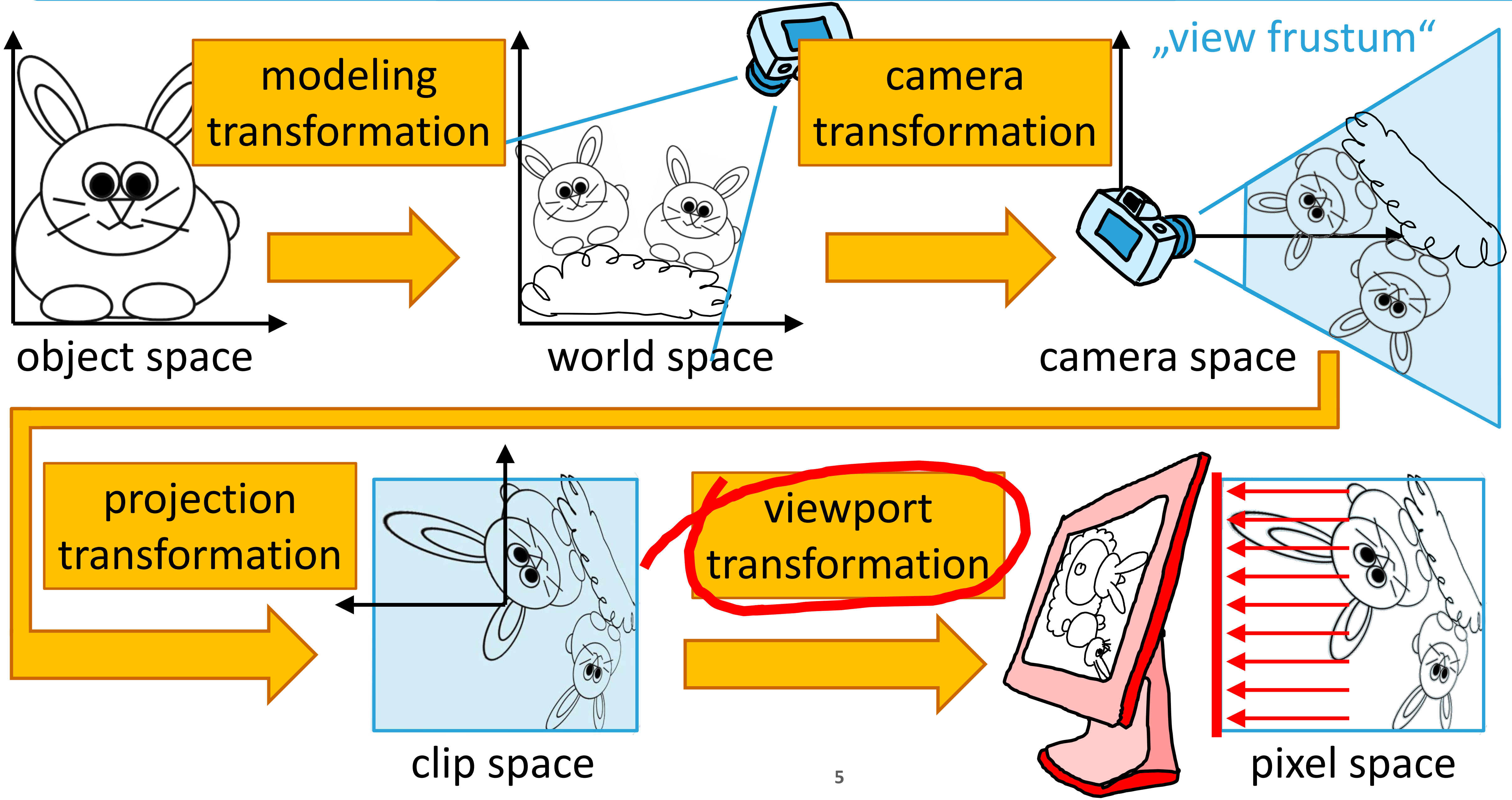


# From Object Space to Screen Space



# Viewport Transformation

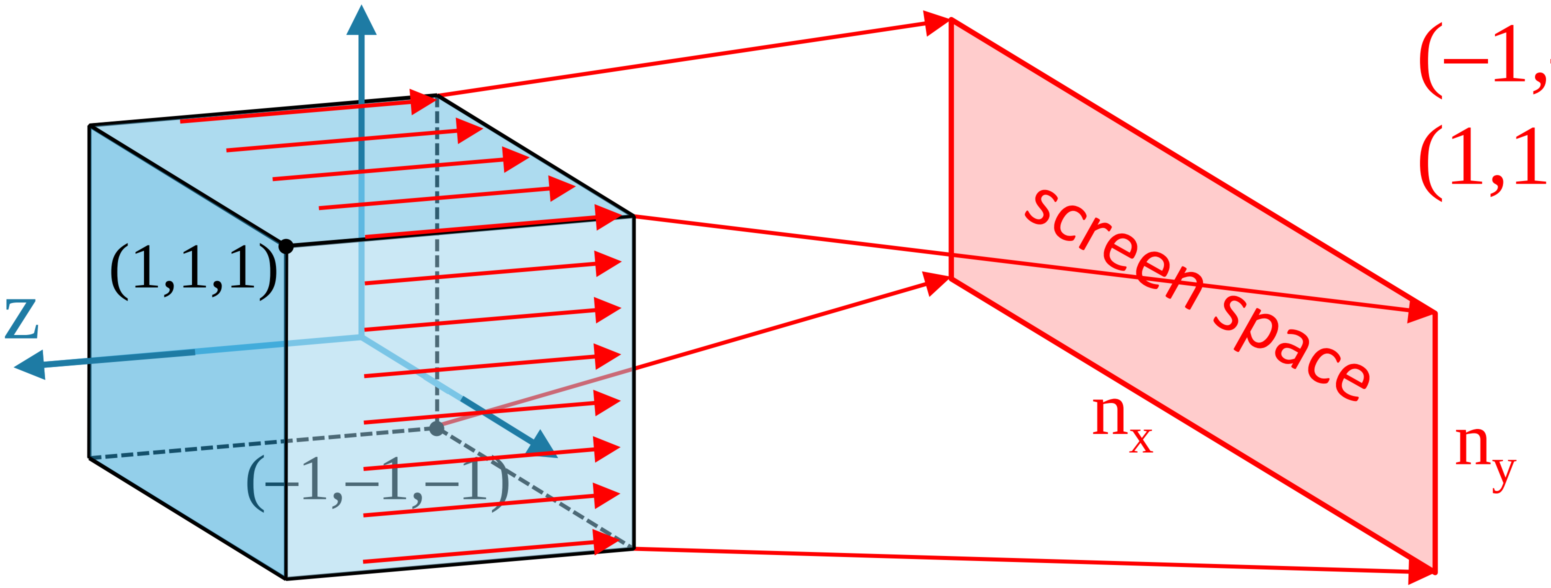
# From Object Space to Screen Space



# Viewport Transformation

- assumption: scene is in clip space !
- clip space = $[-1,1] \times [-1,1] \times [-1,1]$
- orthographic camera looking in $-z$ direction
- screen resolution $n_x \times n_y$ pixels

clip space:




$(-1,-1) \rightarrow (0,0)$

$(1,1) \rightarrow (n_x,n_y)$

# Viewport Transformation

can be done with the matrix

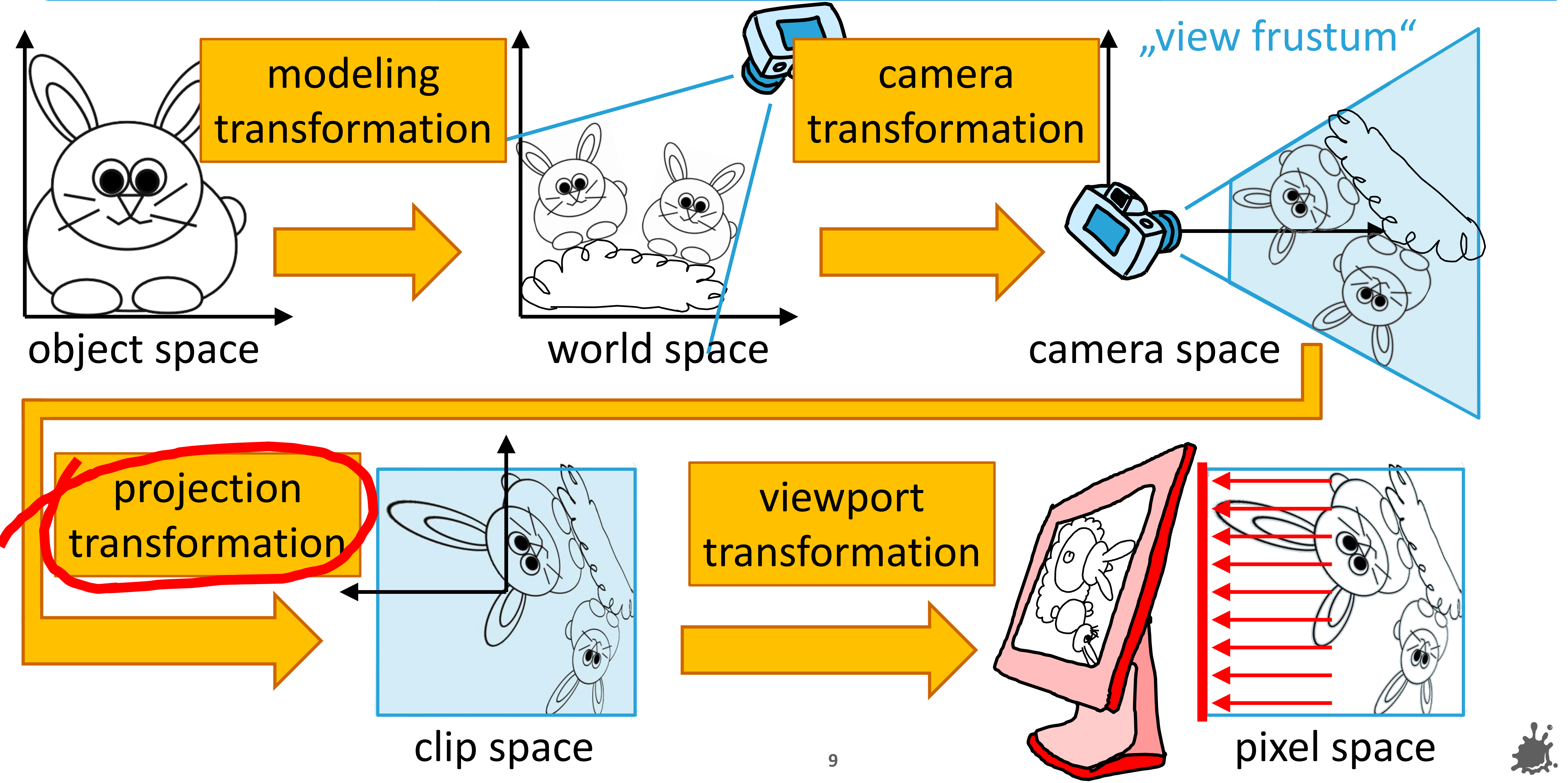
$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ 1 \end{bmatrix} = \begin{bmatrix} n_x/2 & 0 & n_x/2 \\ 0 & n_y/2 & n_y/2 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$(-1,-1) \rightarrow (0,0)$
$(1,1) \rightarrow (n_x,n_y)$

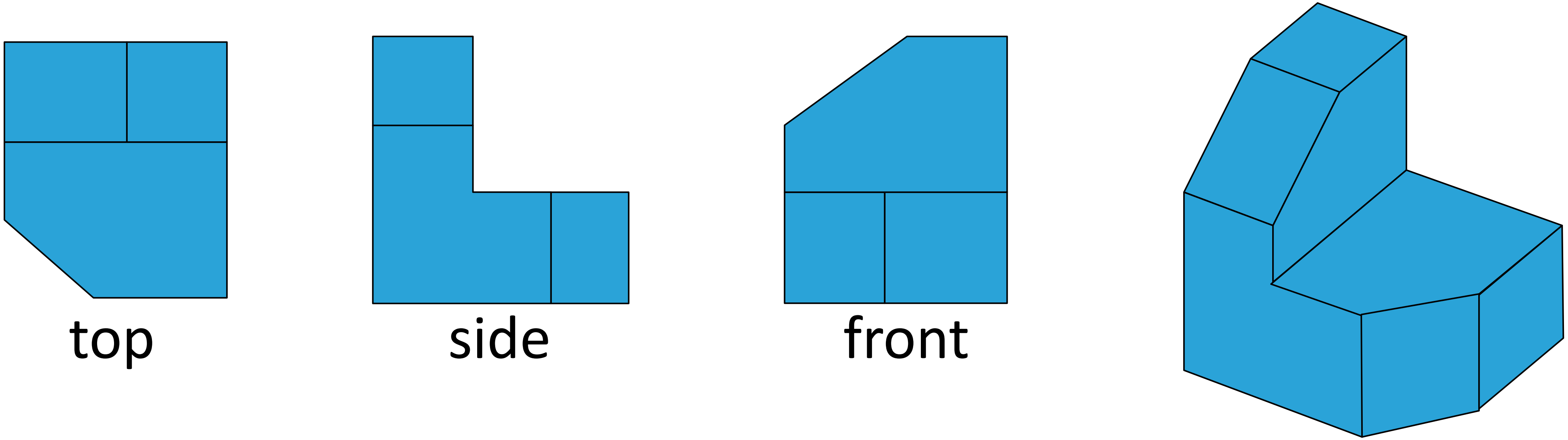
this ignores the z-coordinate, but...

# Projection Transformation

# From Object Space to Screen Space

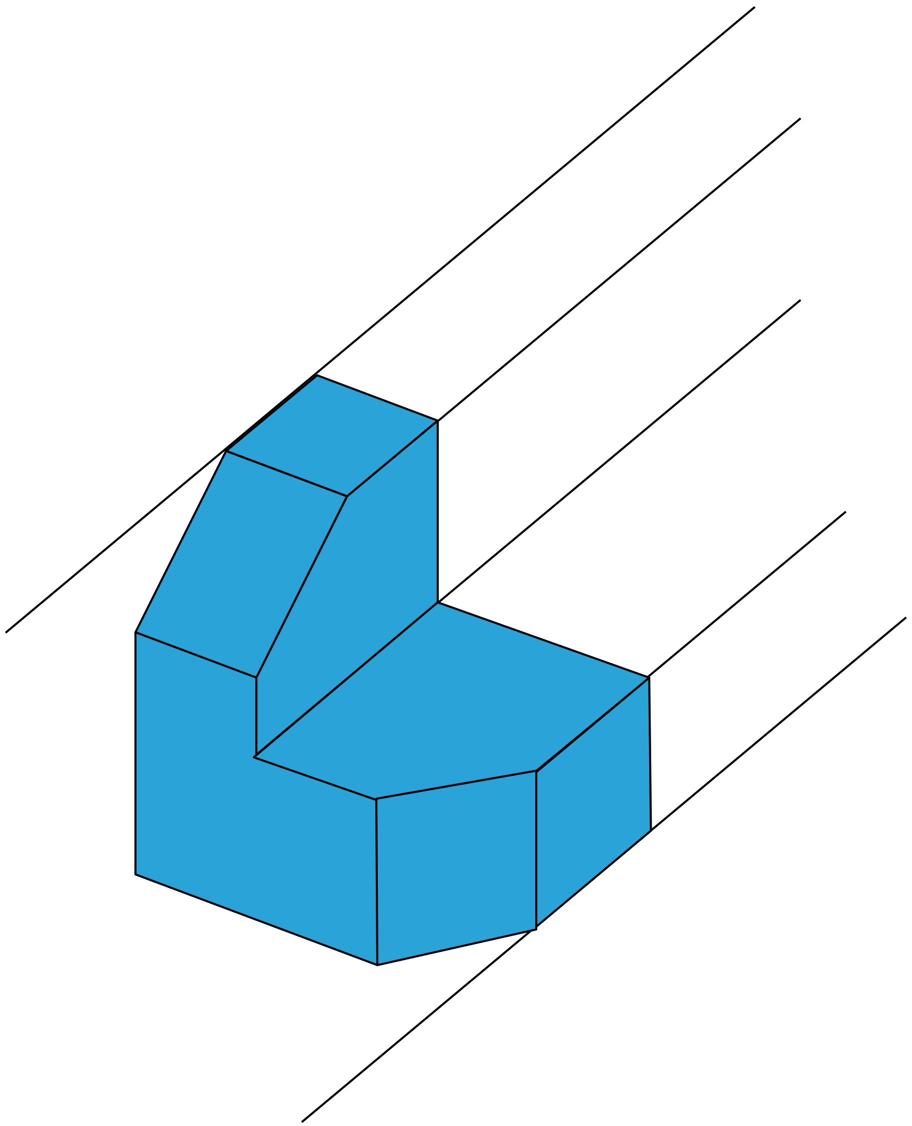


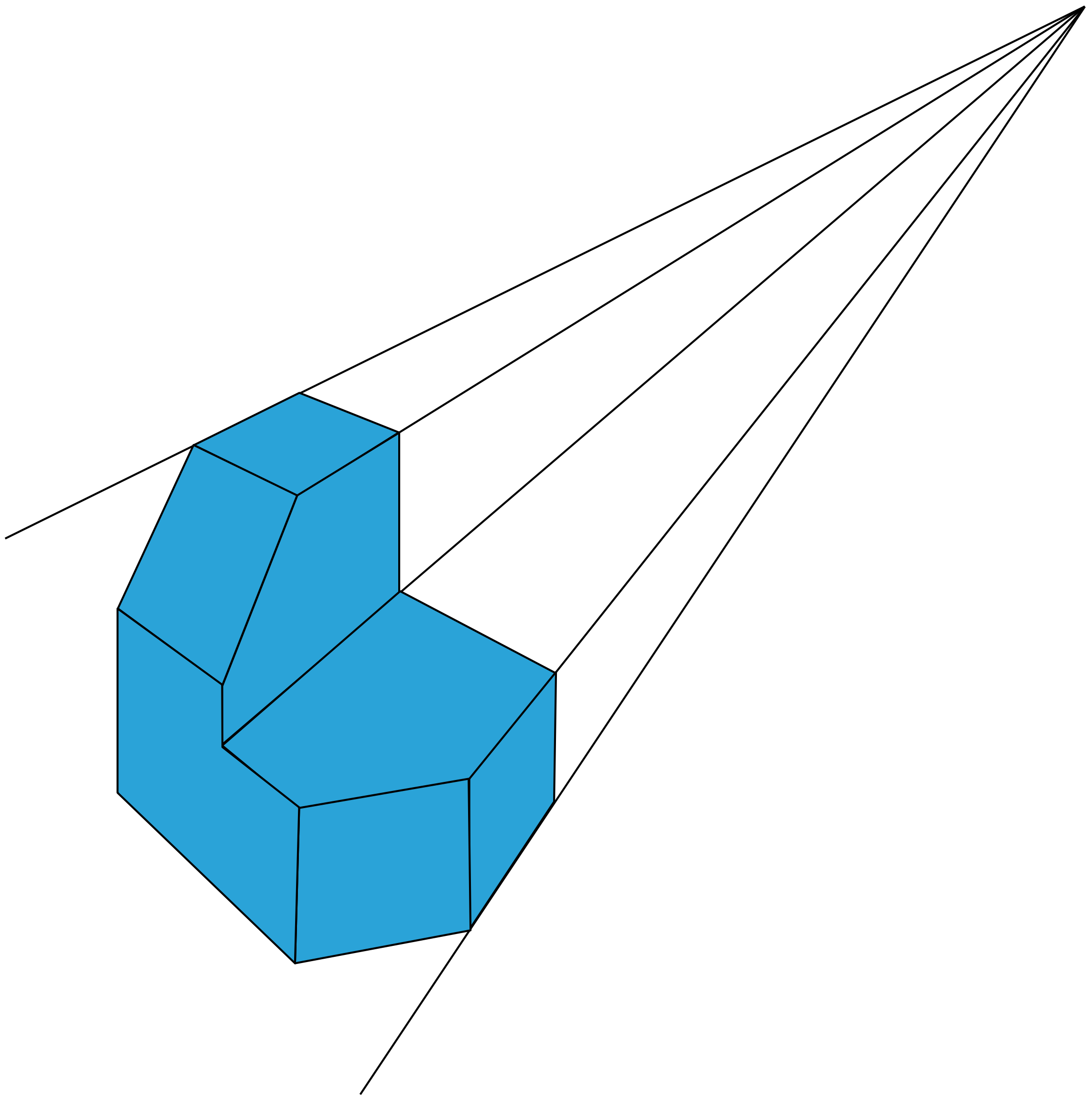
# Parallel Projection (Orthographic Projection)



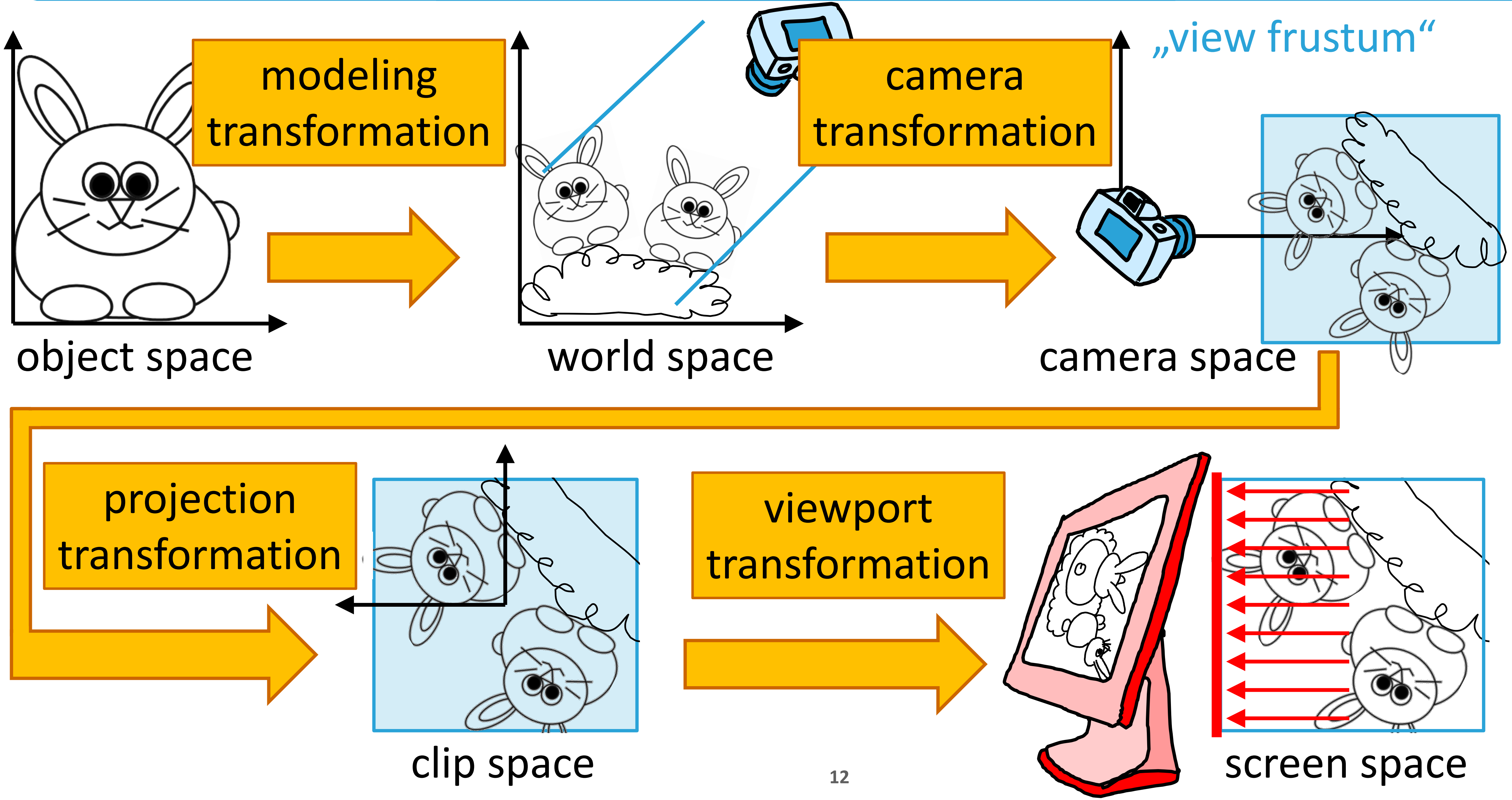

*3 parallel-projection views of an object, showing relative proportions from different viewing positions*

# Parallel vs. Perspective Projection





# For Now: Parallel (Orthographic) Projection



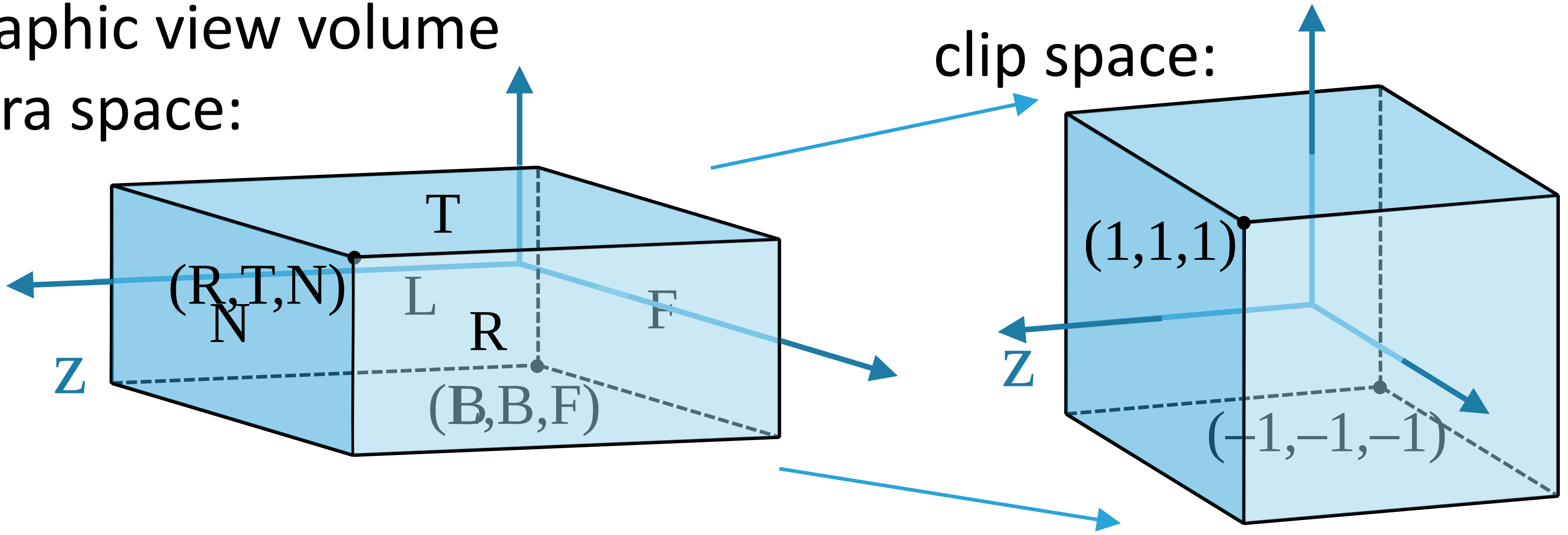
# Projection Transformation (Orthographic)

- assumption: scene in box $[L,R] \times [B,T] \times [F,N]$
- orthographic camera looking in $-z$ direction
- transformation to clip space

$(L,B,F) \rightarrow (-1,-1,-1)$
$(R,T,N) \rightarrow (1,1,1)$

orthographic view volume
in camera space:

clip space:



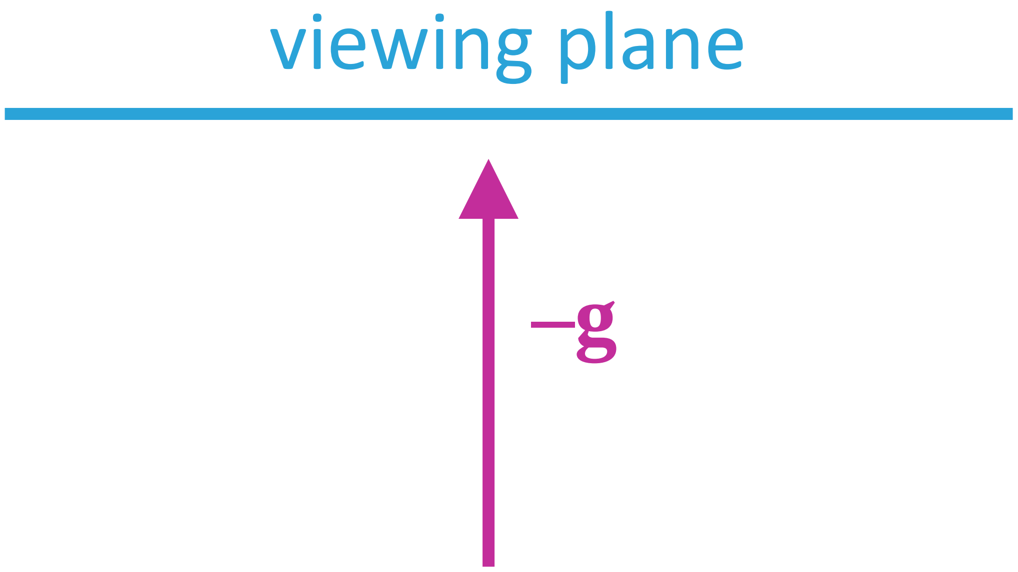
# Projection Transformation (Orthographic)

$(L,B,F) \rightarrow (-1,-1,-1)$
$(R,T,N) \rightarrow (1,1,1)$

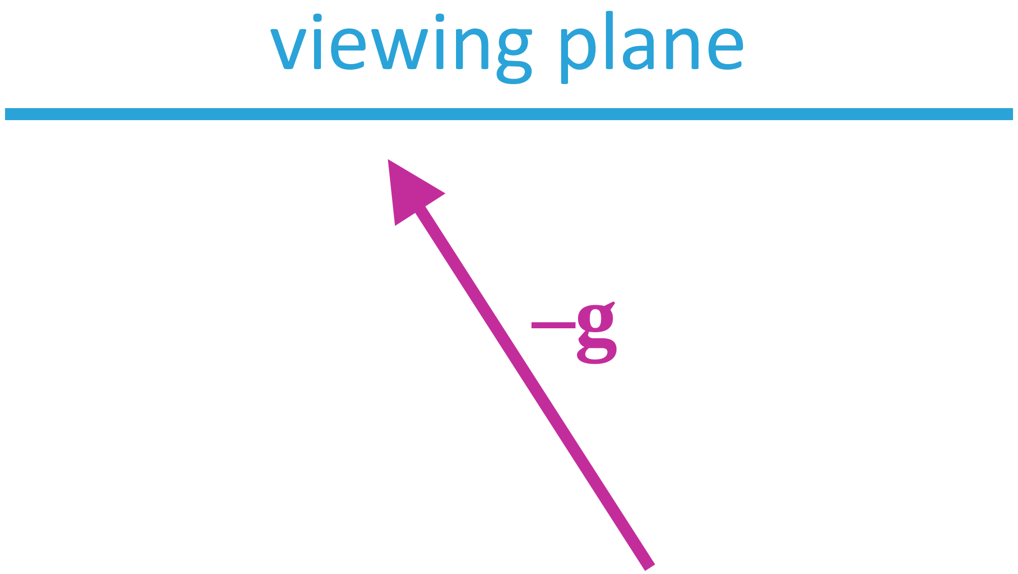
$$\frac{2}{R-L} \cdot L + 0 \cdot B + 0 \cdot F - \frac{R+L}{R-L} \cdot 1 = \frac{L-R}{R-L} = -1$$

$$M_{orth} = \begin{bmatrix} \frac{2}{R-L} & 0 & 0 & -\frac{R+L}{R-L} \\ 0 & \frac{2}{T-B} & 0 & -\frac{T+B}{T-B} \\ 0 & 0 & \frac{2}{N-F} & -\frac{N+F}{N-F} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} L \\ B \\ F \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

# Parallel Projection Types




**orthographic**
projection

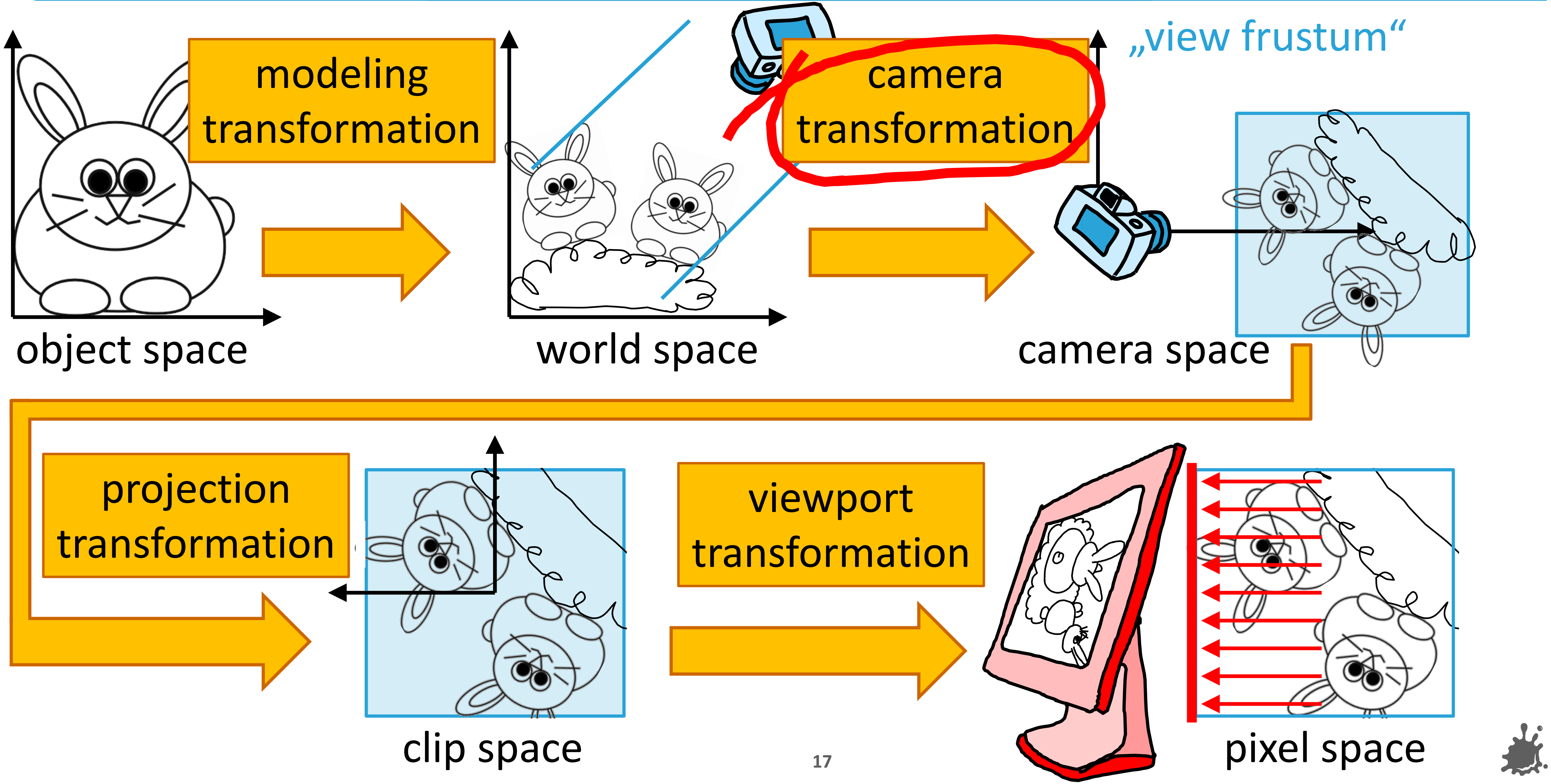



**oblique**
projection

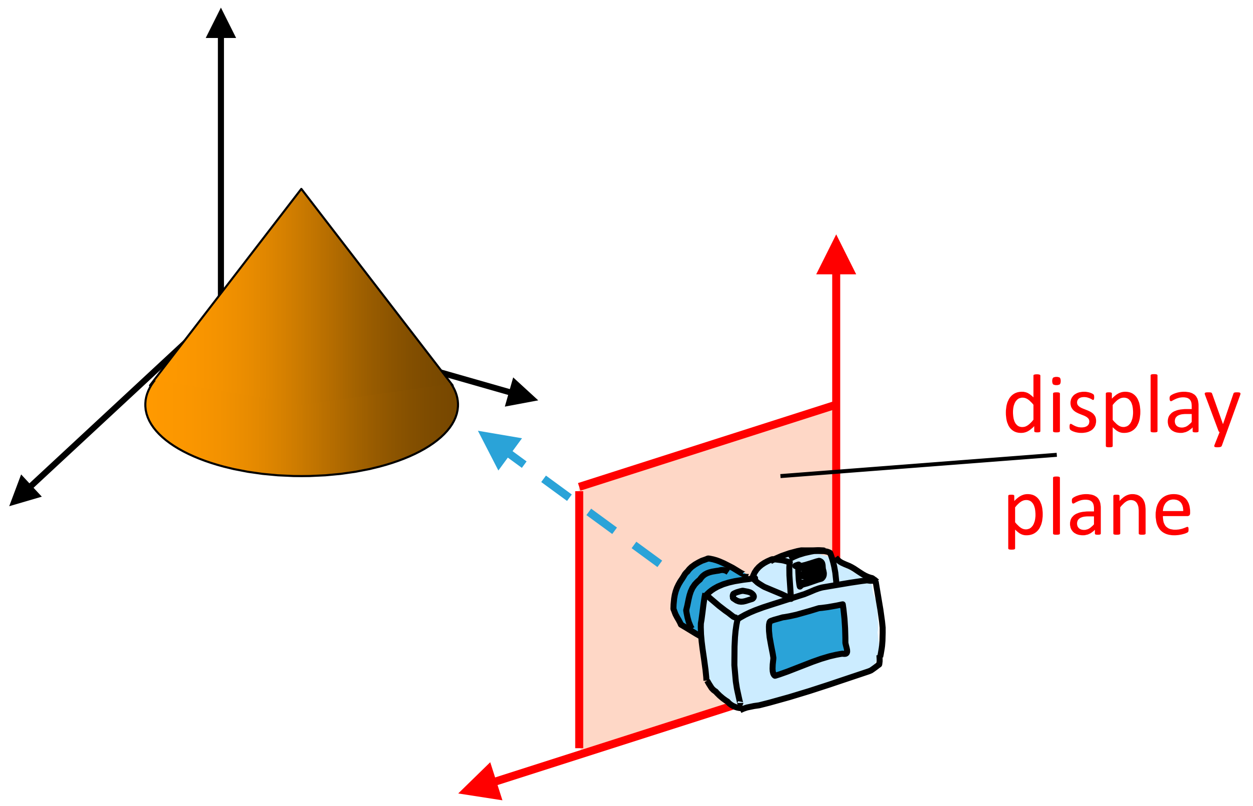
*different orientation of the projection vector –g*

# Camera Transformation

# For Now: Parallel (Orthographic) Projection



# Viewing: Projection Plane

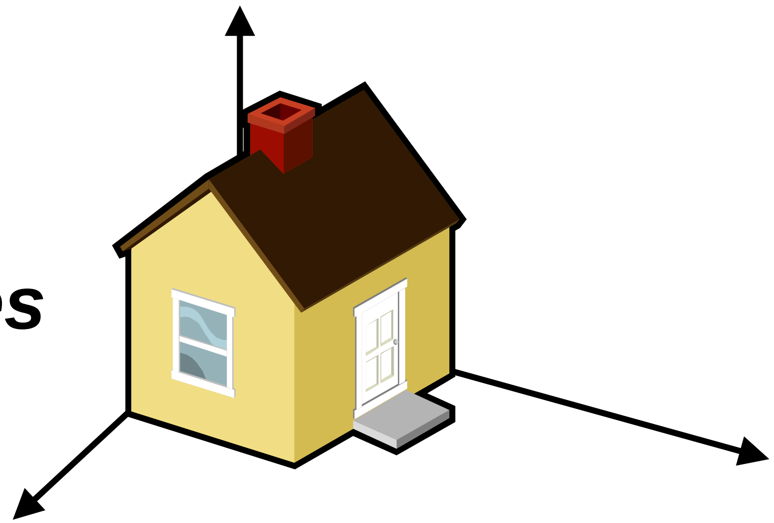


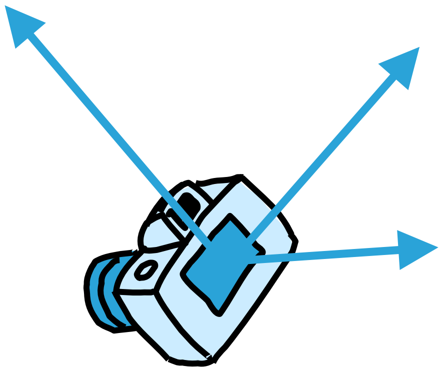

*coordinate reference for obtaining a selected view of a 3D scene*

# Viewing: Camera Definition

- similar to taking a photograph
- involves selection of
  - camera position
  - camera direction
  - camera orientation
  - “window” (zoom) of camera

***world coordinates***

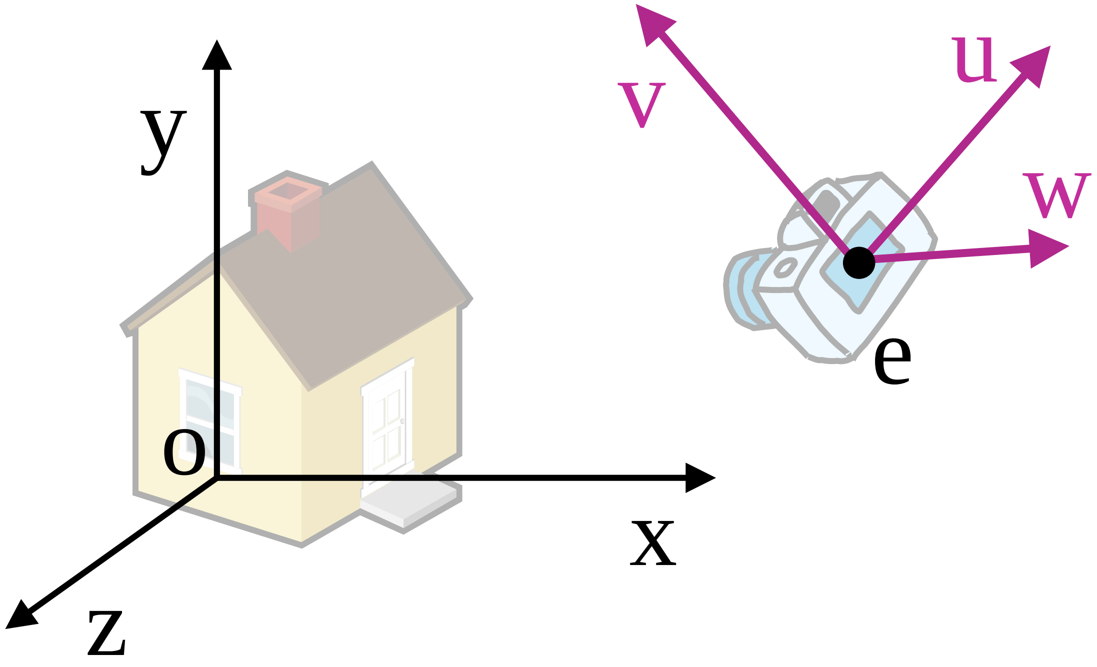




***camera coordinates***

# Viewing: Camera Transformation (1)

- view reference point
  - origin of camera coordinate system
  - gaze direction or look-at point




*right-handed camera-coordinate system,*
*with axes u, v, w, relative to world-coordinate scene*

# Viewing: Camera Transformation (2)

e … eye position

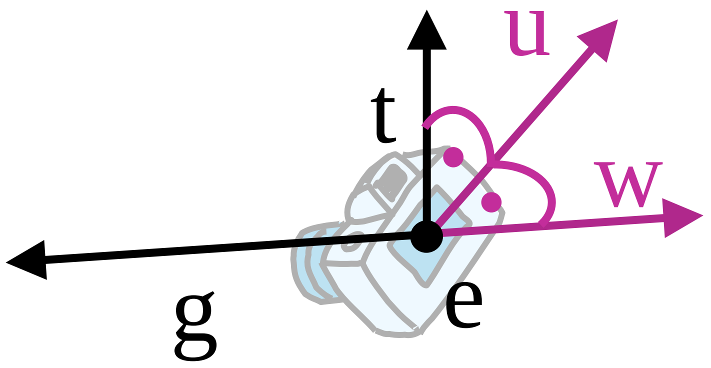
g … gaze direction  (positive w-axis points to the viewer)

t  … view-up vector

$$w = -\frac{g}{|g|}$$

$$u = \frac{t \times w}{|t \times w|}$$

$$v = w \times u$$



# Viewing: Camera Transformation (2)

e … eye position

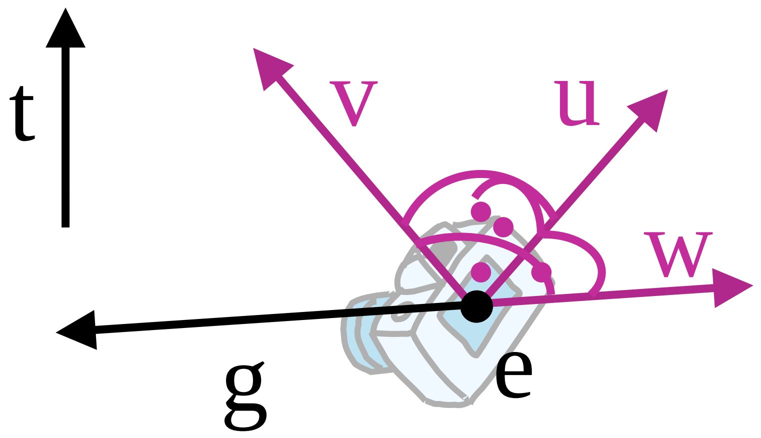
g … gaze direction (positive w-axis points to the viewer)

t … view-up vector

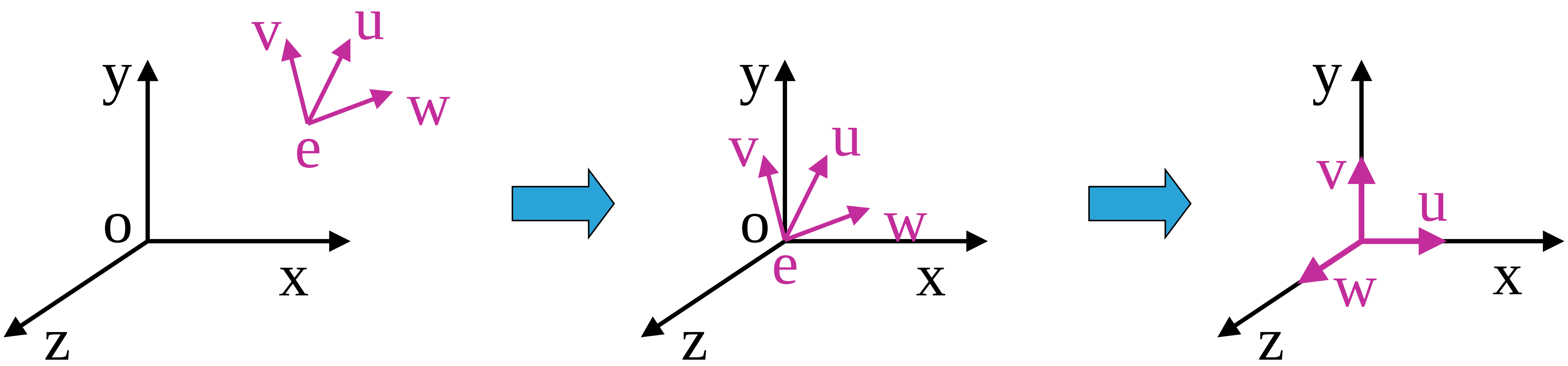
$$w = -\frac{g}{|g|}$$

$$u = \frac{t \times w}{|t \times w|}$$

$$v = w \times u$$



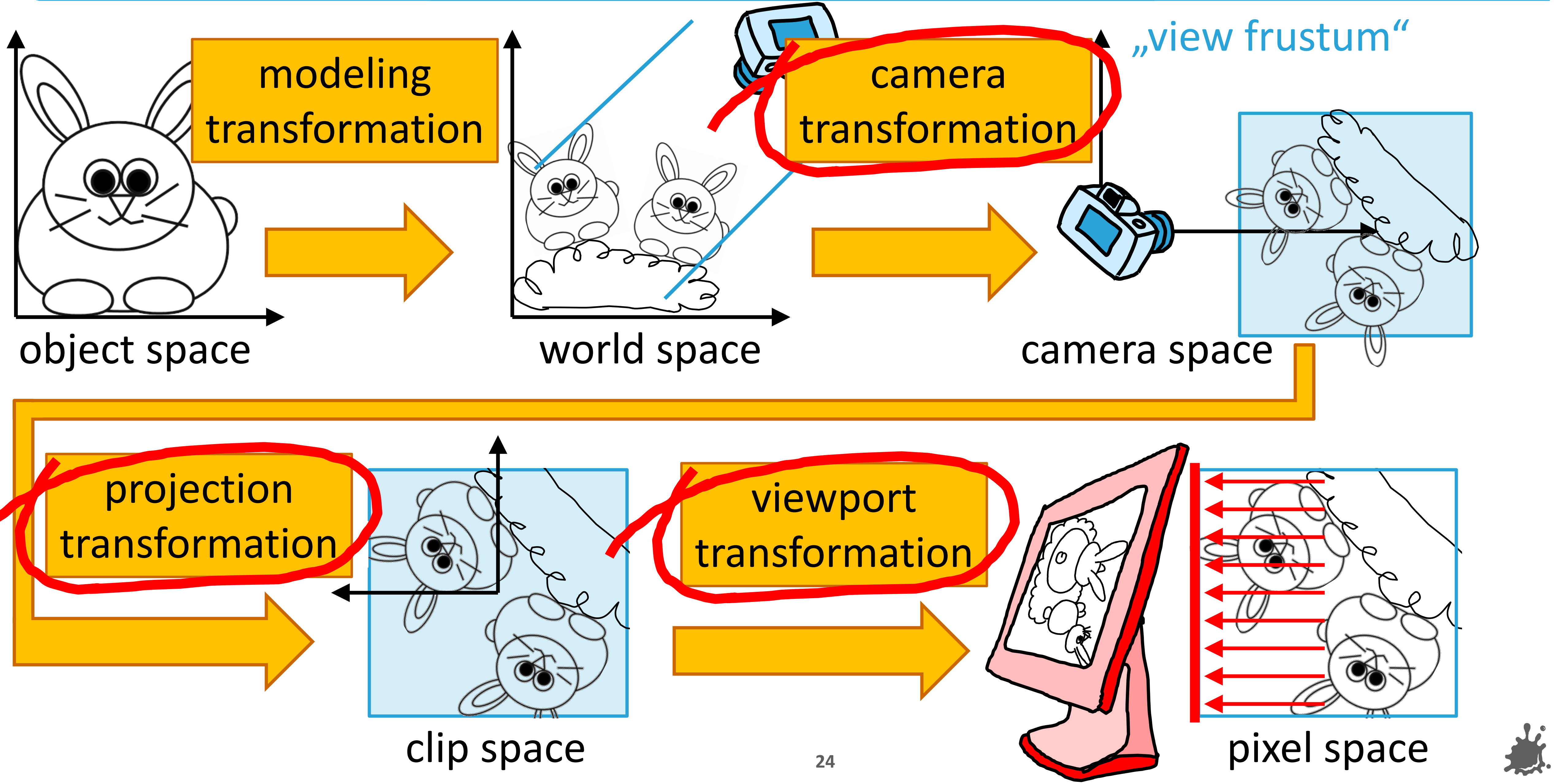
# Viewing: Camera Transformation (3)




$$\mathbf{M_{cam} = R_z \cdot R_y \cdot R_x \cdot T}$$

aligning viewing system with world-coordinate axes
using translate-rotate transformations

# For Now: Parallel (Orthographic) Projection



# Viewing: Camera + Projection + Viewport

$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ z \\ 1 \end{bmatrix} = \left( M_{vp} \cdot M_{orth} \cdot M_{cam} \right) \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

viewport transformation

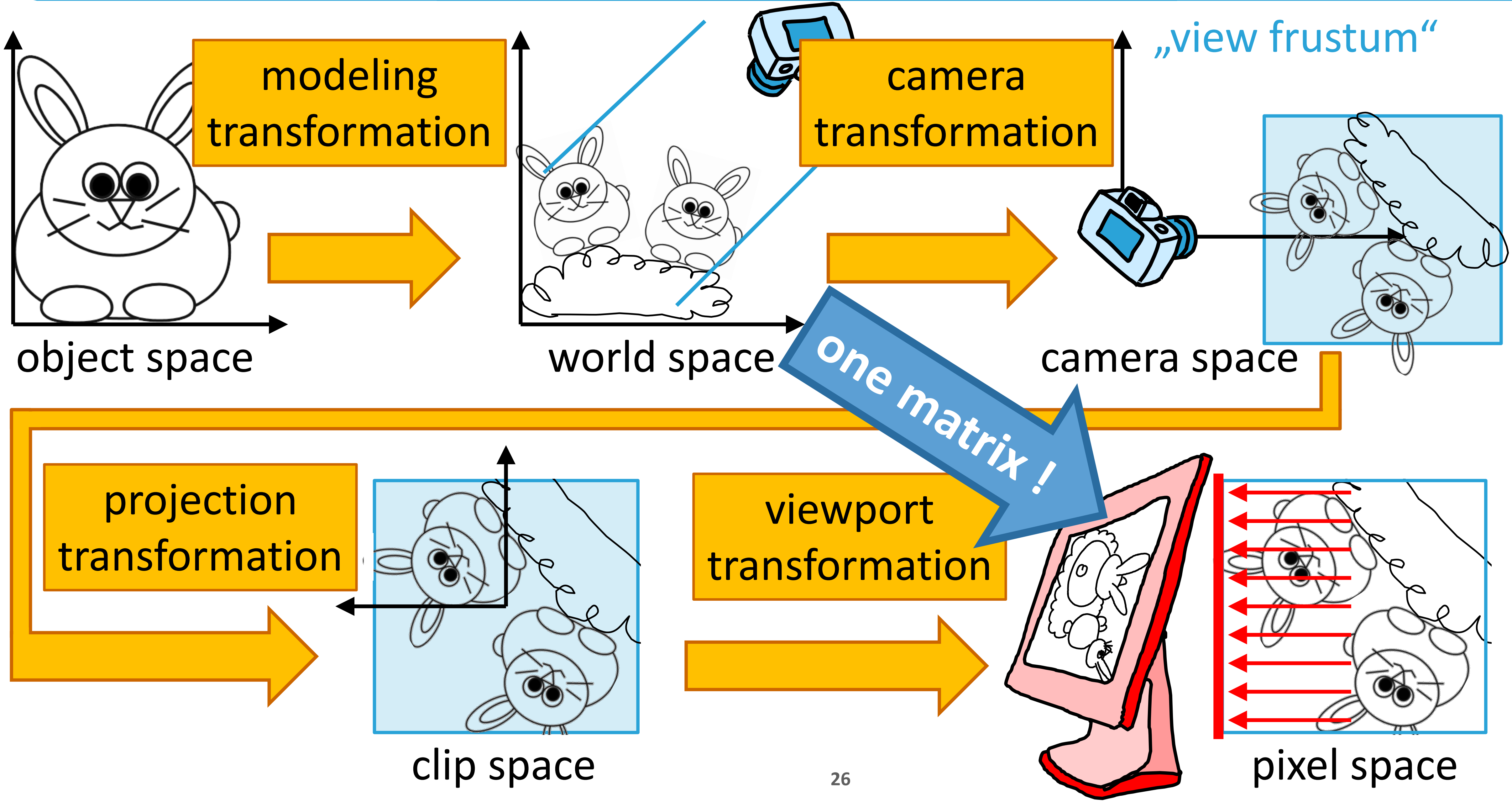
projection transformation

camera transformation

world coordinates

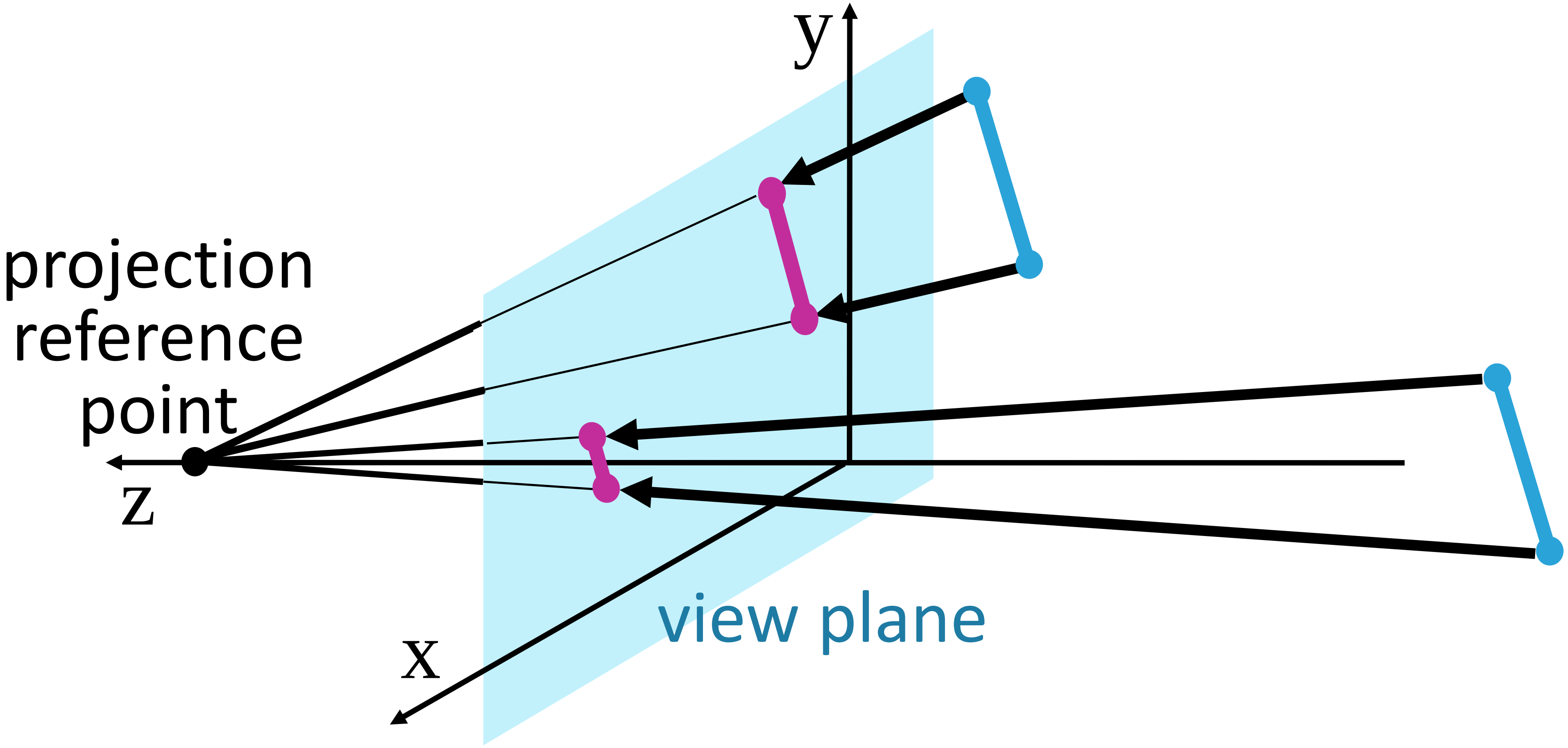
pixels on
the screen

# For Now: Parallel (Orthographic) Projection



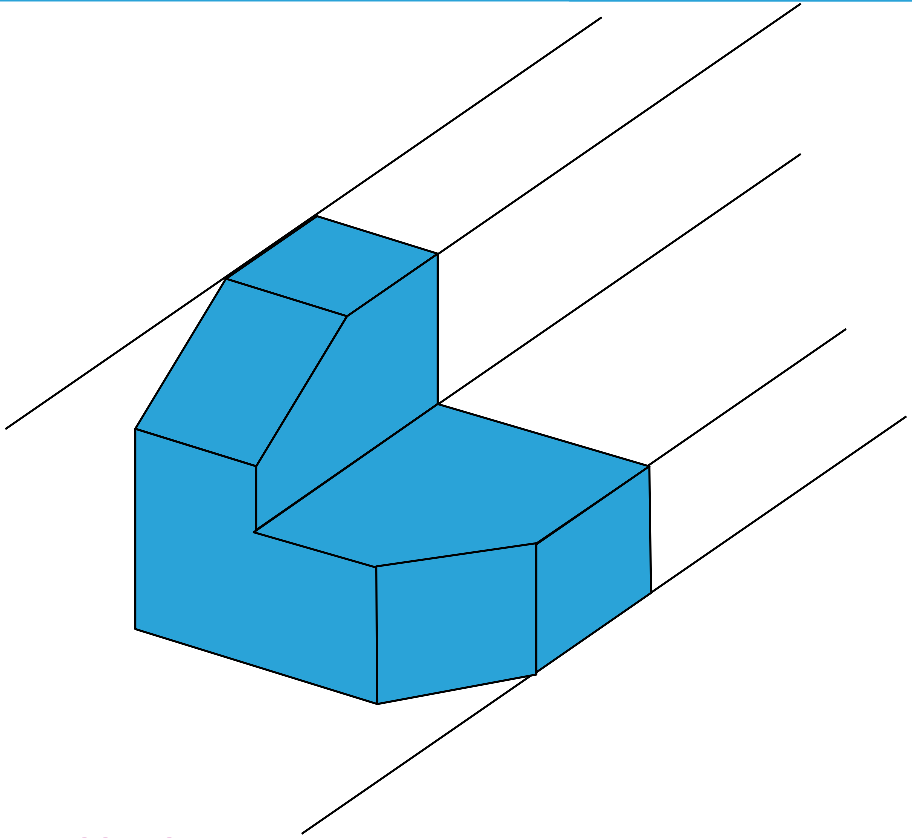
# Perspective Projection

# Perspective Projection

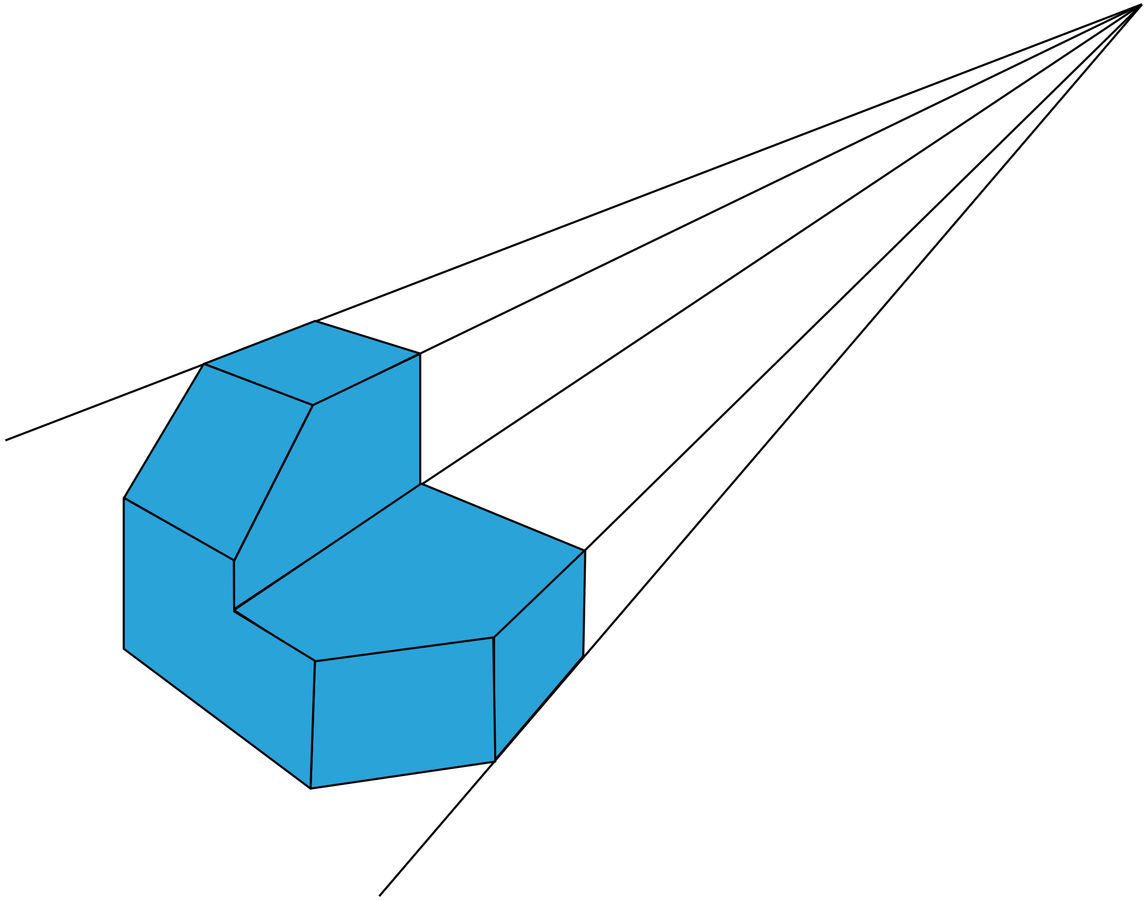



perspective projection of equal-sized objects at different distances from the view plane

# Parallel vs. Perspective Projection

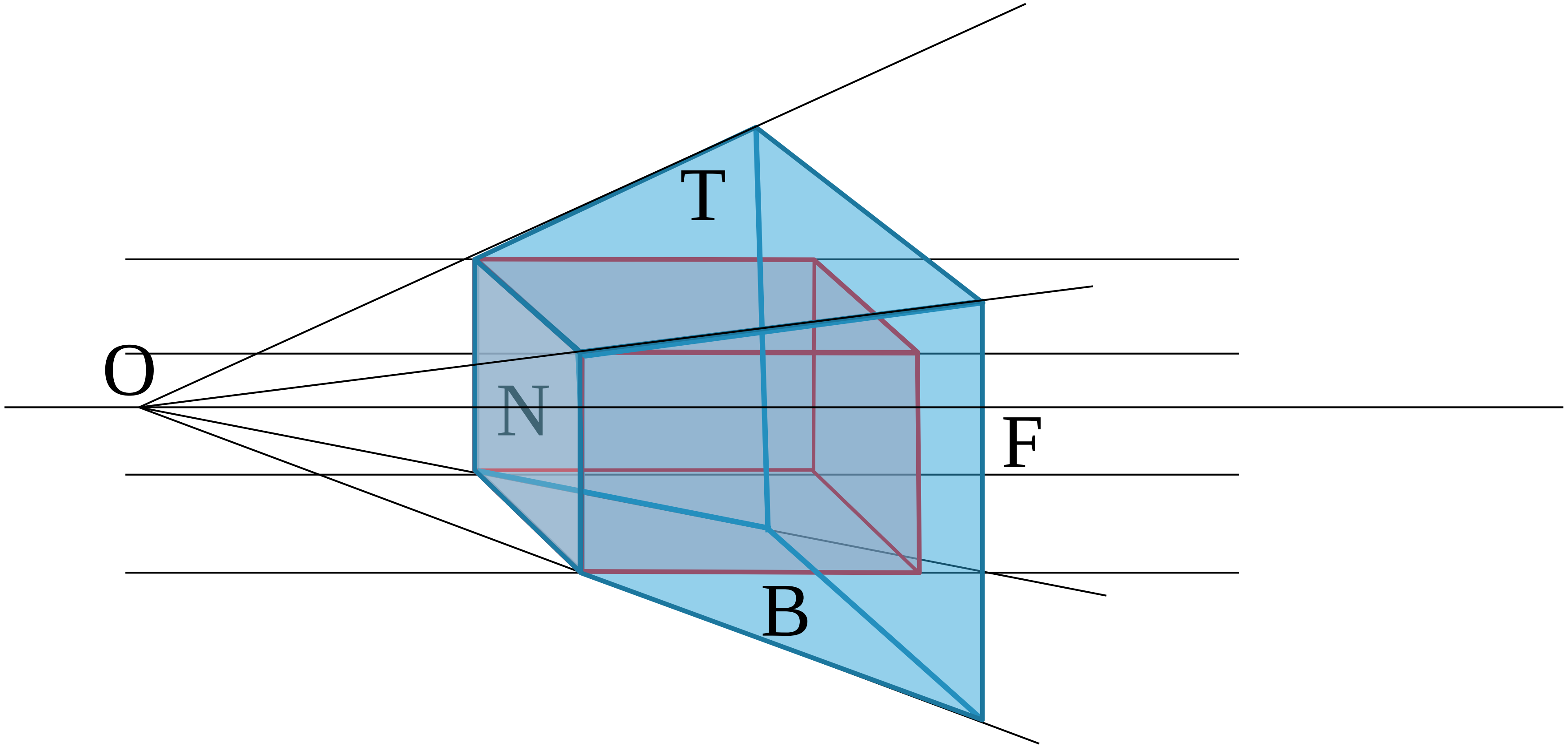


**parallel** projection: preserves relative proportions & parallel features (affine transform.)

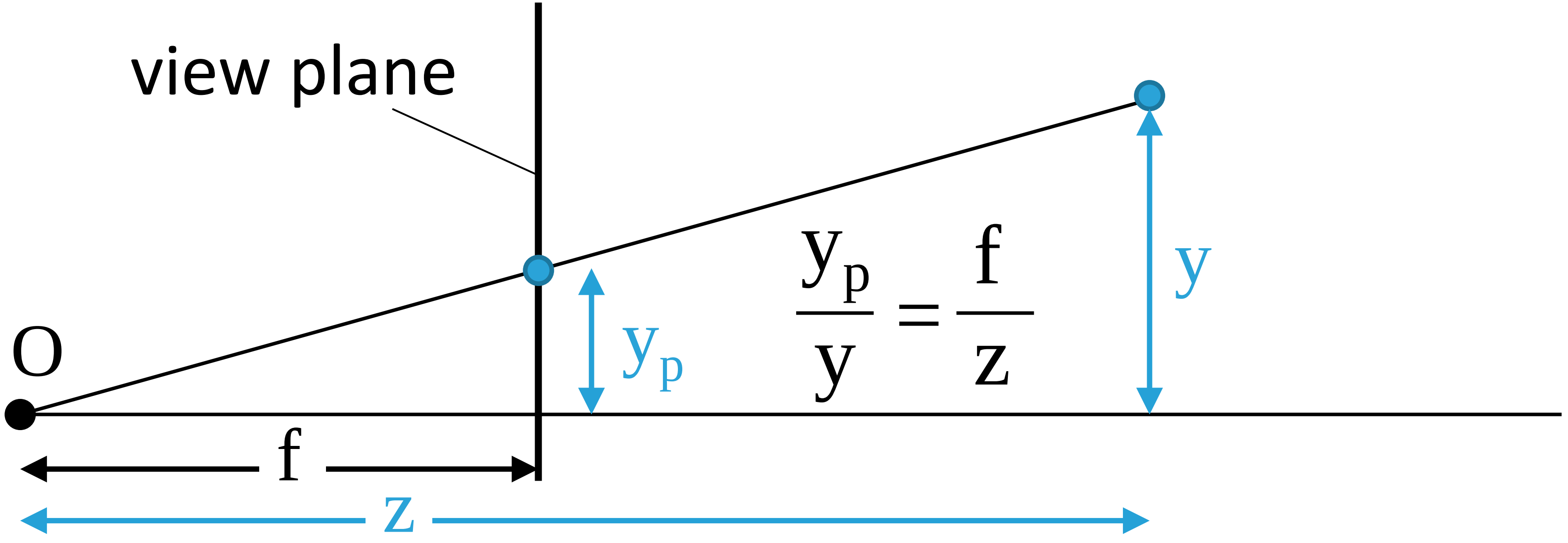


**perspective** projection: center of projection, realistic views

# Perspective Transform



# Perspective Transformation

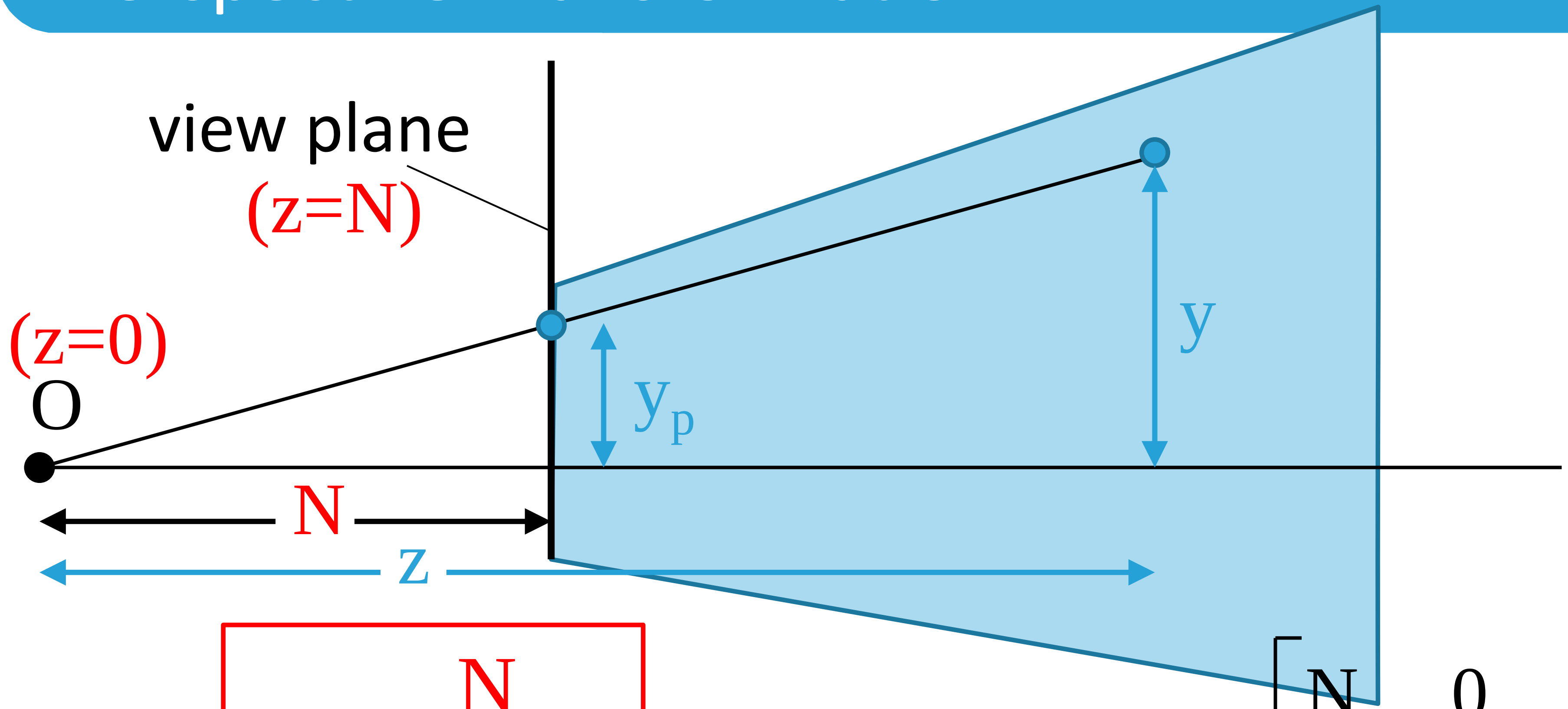



*derivation of perspective transformation*

$$y_p = \frac{f}{z}\, y$$

f … focal length

# Perspective Transformation




*derivation of perspective transformation*

$$y_p = \frac{N}{z}\, y$$

analogous:

$$x_p = \frac{N}{z}\, x$$

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathbf{P}$$

# Perspective Transformation

$$\mathbf{P} = \begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F \cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \cdot N \\ y \cdot N \\ z \cdot (N+F) - F \cdot N \\ z \end{bmatrix}$$

*derivation of perspective transformation*

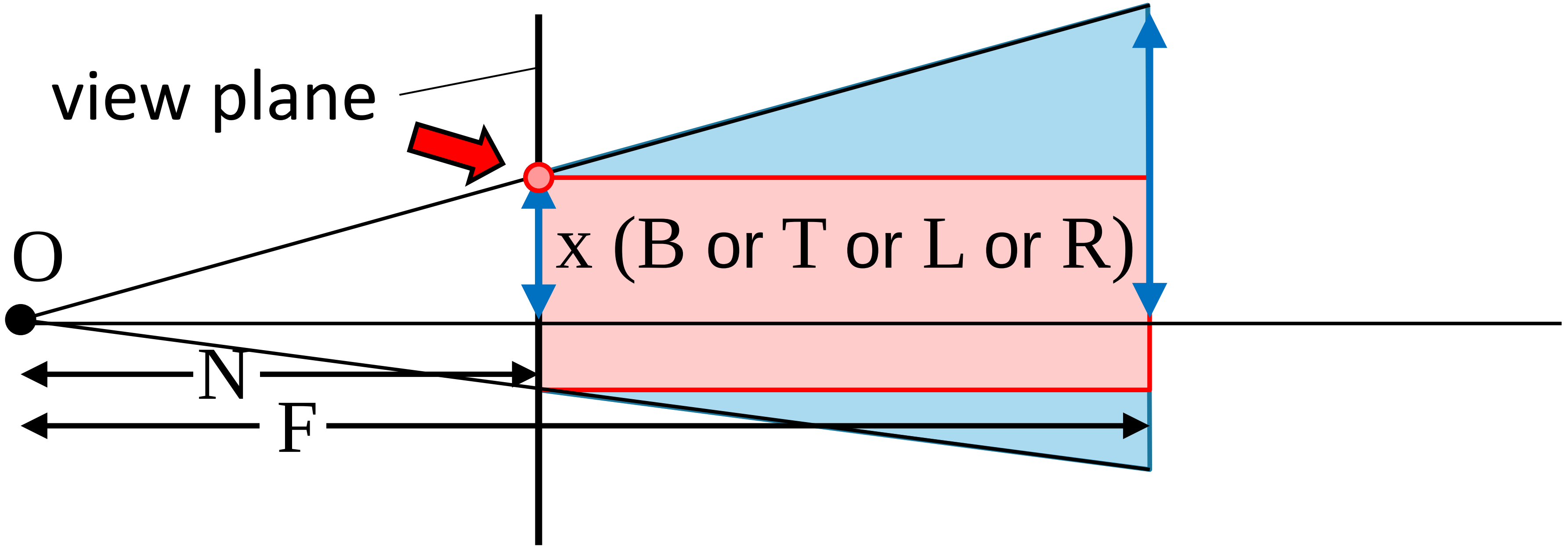
homogenization: divide by z

$$\boxed{\begin{aligned} y_p &= \frac{N}{z}\, y \\ x_p &= \frac{N}{z}\, x \end{aligned}}$$

$$\sim> \begin{bmatrix} x \cdot N/z \\ y \cdot N/z \\ (N+F) - F \cdot N/z \\ 1 \end{bmatrix}$$

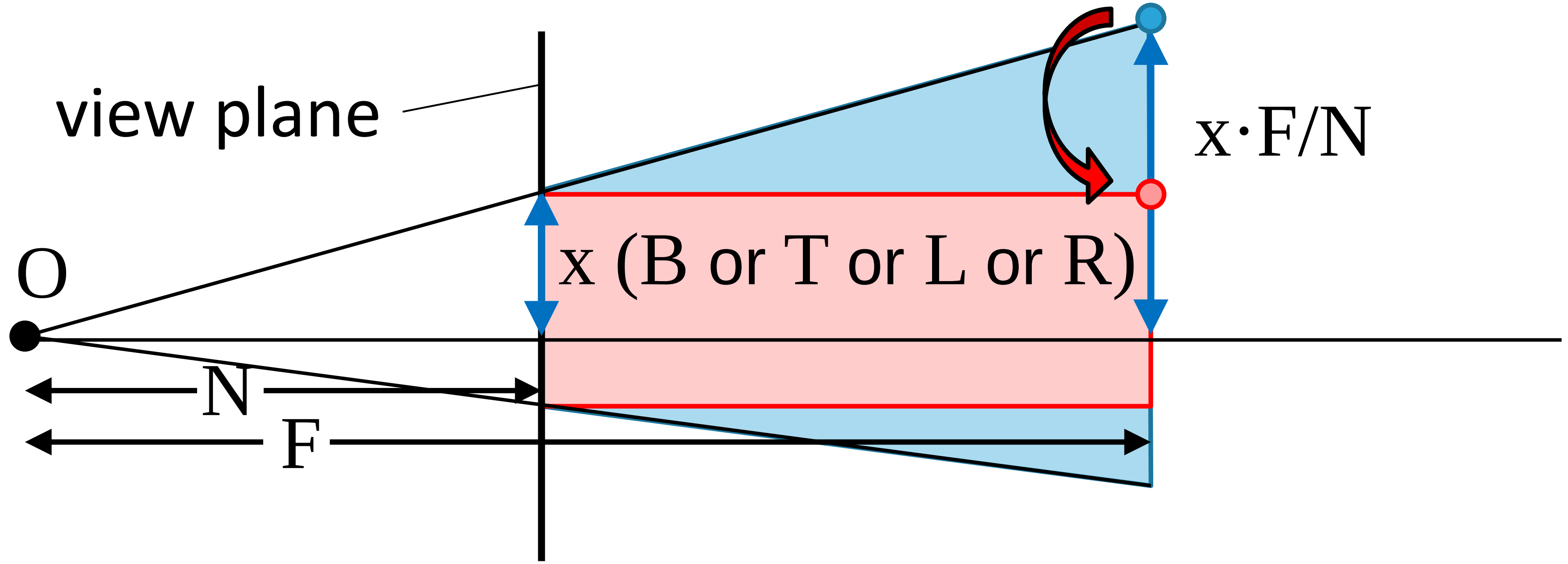
# Example: (Right) Top Near Corner

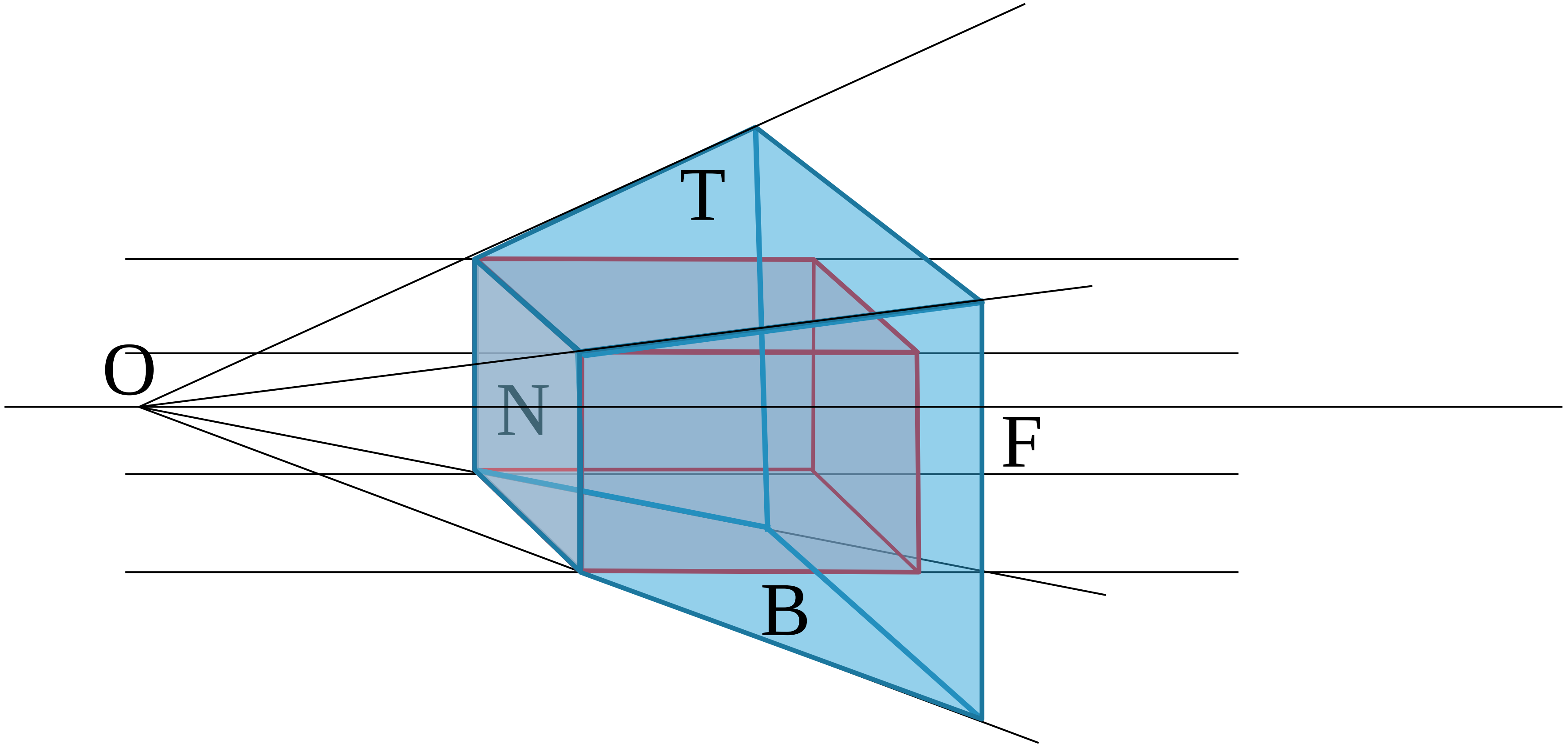
$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R \\ T \\ N \\ 1 \end{bmatrix} = \begin{bmatrix} R\cdot N \\ T\cdot N \\ N\cdot(N+F) - F\cdot N \\ N \end{bmatrix} \sim> \begin{bmatrix} R \\ T \\ N \\ 1 \end{bmatrix}$$



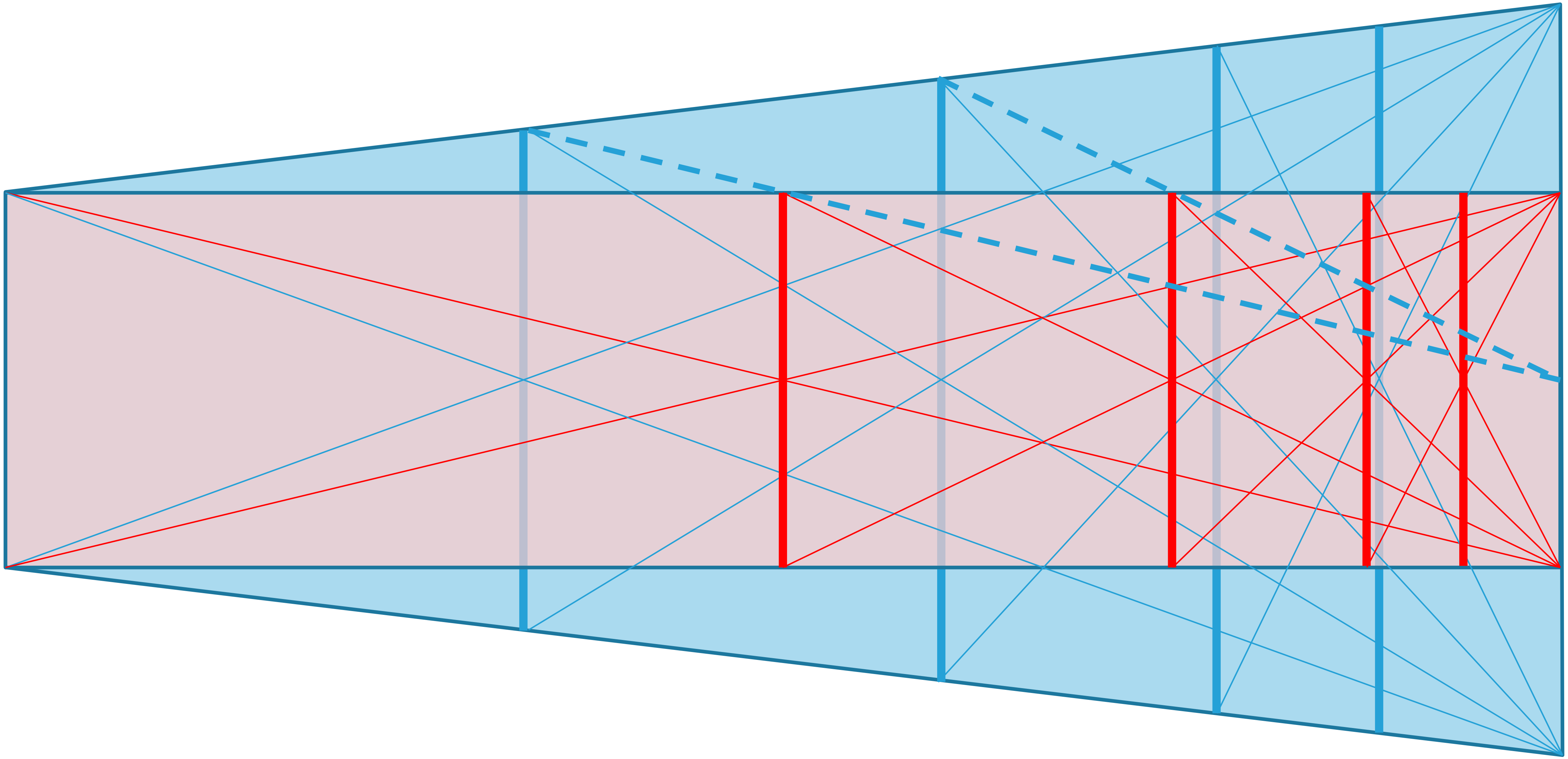
# Example: (Left) Top Far Corner

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} L\cdot F/N \\ T\cdot F/N \\ F \\ 1 \end{bmatrix} = \begin{bmatrix} L\cdot F \\ T\cdot F \\ F\cdot(N+F) - F\cdot N \\ F \end{bmatrix} \sim> \begin{bmatrix} L \\ T \\ F \\ 1 \end{bmatrix}$$

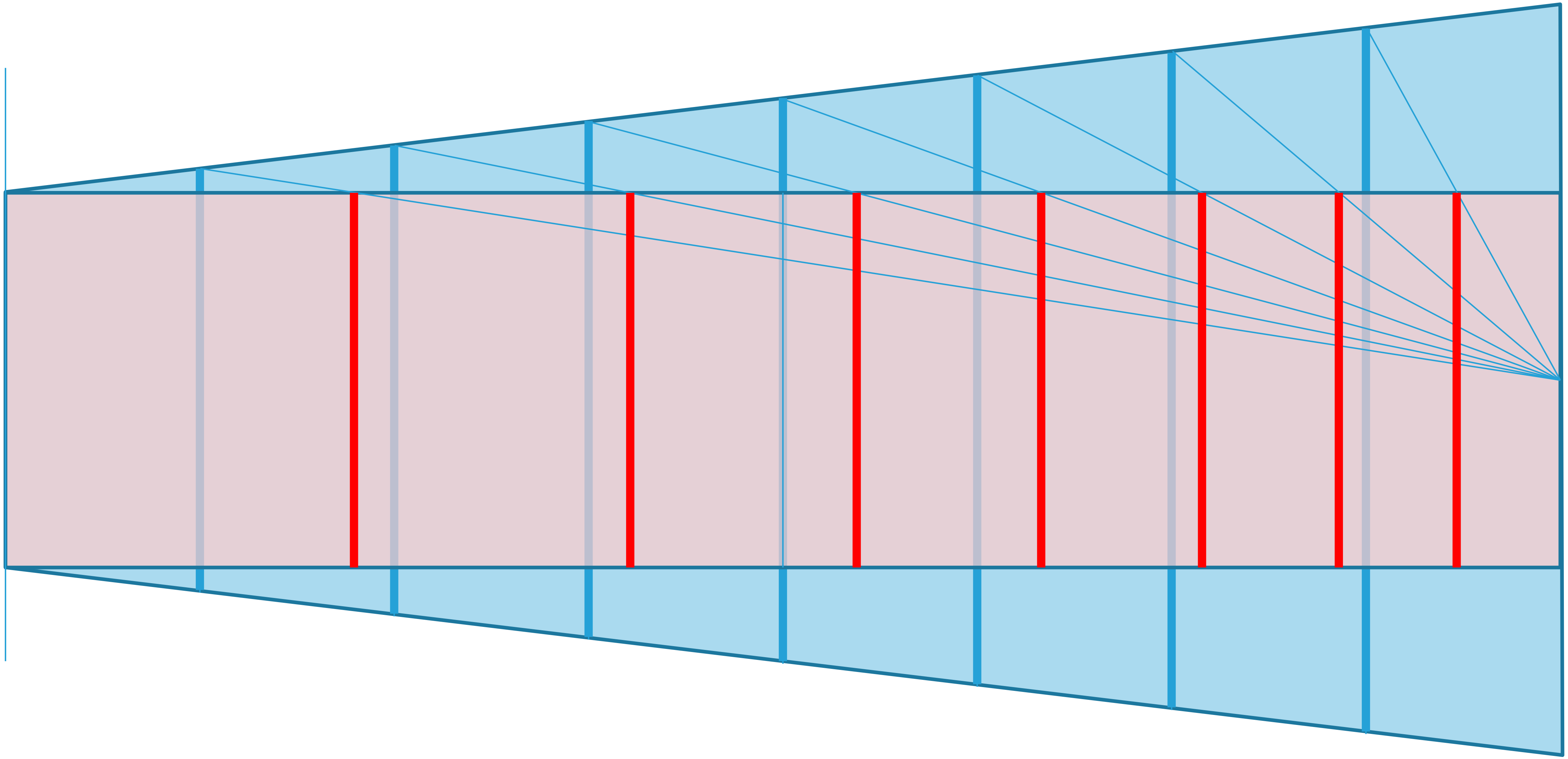



T
O
N
F
B

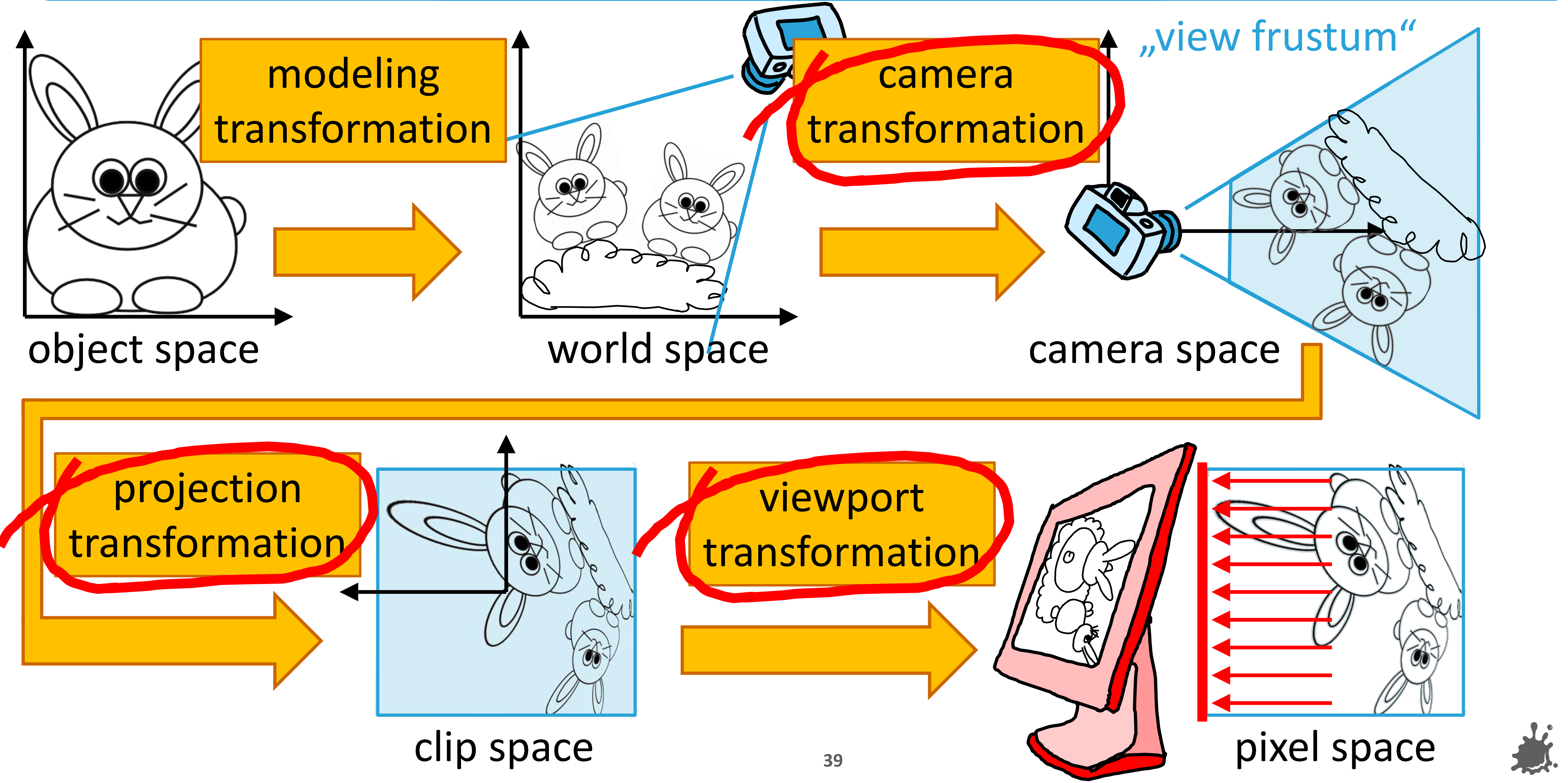
# Nonlinear z-Behaviour



# Nonlinear z-Behaviour



# From Object Space to Screen Space



# Viewing: Camera + Projection + Viewport

$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ z' \\ 1 \end{bmatrix} = M_{vp} \cdot \left( \overbrace{M_{orth} \cdot P}^{M_{per}} \cdot M_{cam} \cdot M_{mod} \right) \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

pixels on the screen

viewport transformation

homogenization & clipping

projection transformation

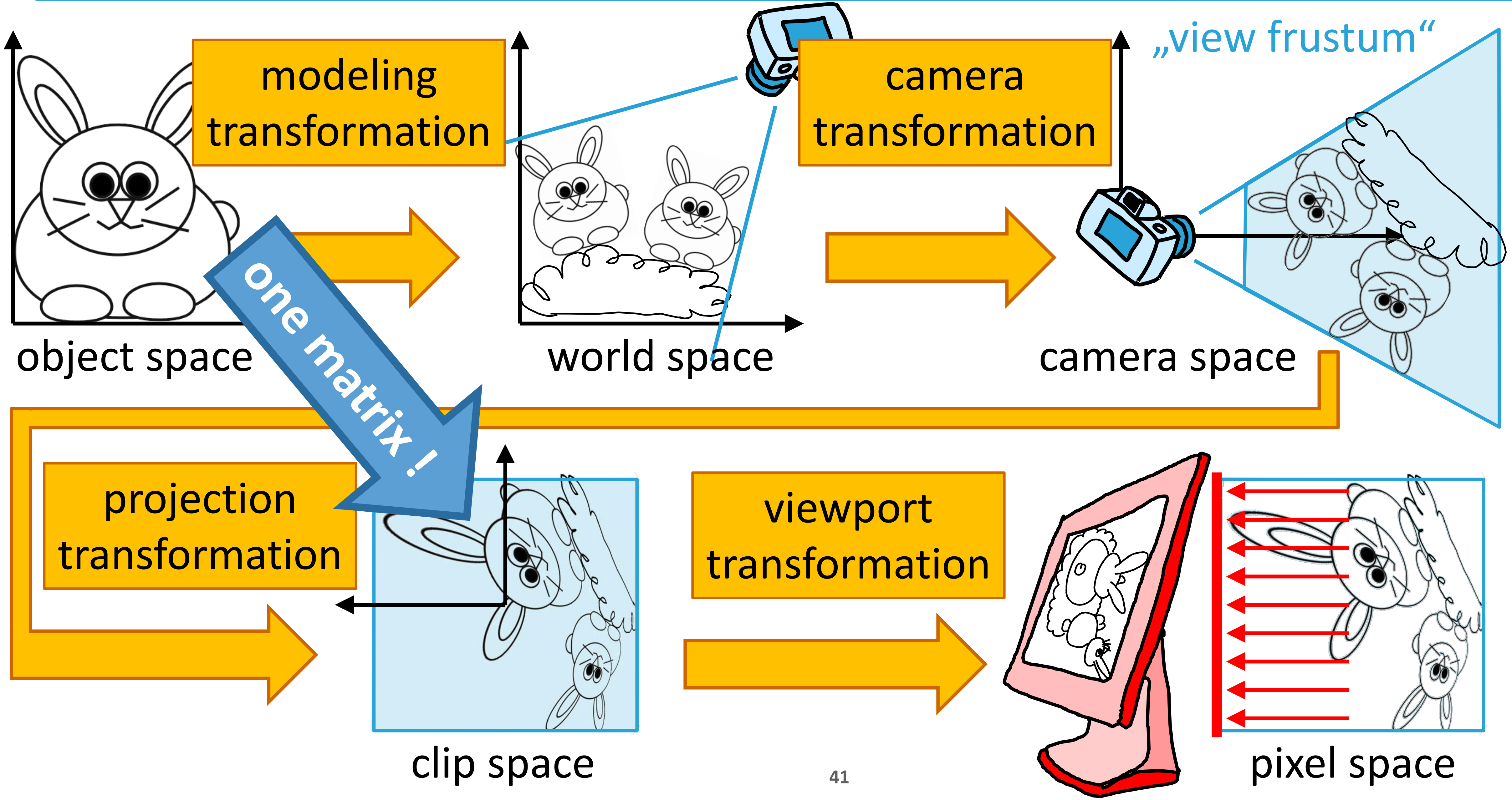
perspective projection

camera transformation

modeling transformation

object coordinates

# From Object Space to Screen Space



# z-Values Remain in Order

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \sim> \begin{bmatrix} x\cdot N/z \\ y\cdot N/z \\ (N+F) - F\cdot N/z \\ 1 \end{bmatrix}$$

$z_1, z_2, N, F < 0$

$z_1 < z_2$

$1/z_1 > 1/z_2 \qquad | \cdot (-F\cdot N) \qquad (<0)$

$-F\cdot N/z_1 < -F\cdot N/z_2 \qquad | + (N+F)$

$(N+F) - F\cdot N/z_1 < (N+F) - F\cdot N/z_2$

# Perspective Projection Properties

- parallel lines parallel to view plane $\Rightarrow$ parallel lines
- parallel lines not parallel to view plane $\Rightarrow$ converging lines (vanishing point)
- lines parallel to coordinate axis $\Rightarrow$
principal vanishing point (one, two or three)

# Principle Vanishing Points